

Candidata: Gleiciane Galdino

Vaga: Estágio em IA

Relatório Técnico: Orquestração de IA com n8n para Análise de Dados

O objetivo desse projeto é desenvolver uma Prova de Conceito (PoC) de um sistema de Business Intelligence, no qual o n8n centraliza a orquestração da migração de dados de planilhas para um banco de dados e atua como motor de uma aplicação de IA conversacional, permitindo que o usuário faça perguntas em linguagem natural e receba respostas precisas baseadas nos dados.

Obtendo a estrutura do projeto:

Ao clonar ou fazer o fork do repositório, você encontrará a seguinte estrutura de arquivos:

/Infraestrutura-Base

- |— Dockerfile (Instruções para construir a imagem da API)
- |— docker-compose.yml (Inicia a API, o banco de dados e o n8n)
- |— main.py (Código da API em Python)
- |— requirements.txt (Dependências do Python)

Acessando a pasta do projeto pelo terminal:

Para rodar o projeto, você precisa primeiro abrir um terminal ou PowerShell dentro da pasta que contém o arquivo **docker-compose.yml**.

Existem duas formas fáceis de fazer isso:

- Usando o comando `cd`: No terminal, digite `cd` (que significa "mudar diretório") seguido do caminho da sua pasta e pressione Enter. Por exemplo: `cd C:\Users\SeuUsuario\Desktop\Projetos\Infraestrutura-Base`
- Direto do Visual Studio Code: Se você já usa o VS Code, pode abrir o terminal diretamente na pasta do projeto. No menu superior, vá em Terminal e selecione New Terminal. O terminal já abrirá no local certo, pronto para você usar.

Como rodar o projeto:

Primeiro deve abrir o docker desktop e acessar a conta cadastrada; o próximo passo é acessar a pasta do projeto (explicado anteriormente), chamada **Infraestrutura-Base**, que contém o arquivo **docker-compose.yml**, abrir um terminal e executar o comando

`docker-compose up -d`. O `docker-compose.yml` está configurado para iniciar a API, o banco de dados e o n8n de uma vez, e eles continuarão rodando em segundo plano.

O comando **docker-compose up**: vai ler o arquivo **docker-compose.yml** e iniciar todos os serviços definidos nele:

- Construindo a imagem da API;
- Baixar a imagem do banco de dados e do n8n;
- Iniciar todos os containers.

A flag **-d** significa "detach", o que faz com que os containers rodem em segundo plano, permitindo o uso contínuo do terminal.

Ou seja:

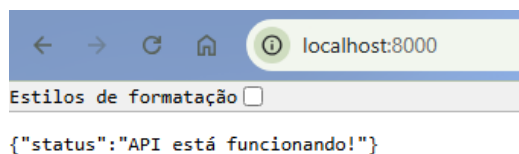
1. Abrir o Docker Desktop e acessar a conta cadastrada;
2. Acessar a pasta do projeto Infraestrutura-Base;
3. Executar o comando: `docker-compose up -d`.

Como saber se deu certo:

Para verificar o status dos containers execute o comando: **docker ps**, a saída mostrará os containers **api_container**, **postgres_container** e **n8n_container** listados e com o status **Up**.

Acessando os Serviços:

- **API**: Estará disponível na porta **8000**.
Pode testá-la acessando <http://localhost:8000> no seu navegador. A resposta deve ser `{"status": "API está funcionando!"}` como na imagem abaixo:



- **n8n**: A instância do n8n estará disponível na porta **5678**. Acesse <http://localhost:5678> no seu navegador e faça login com as credenciais padrão: usuário **admingle** e senha **admingle123**.

- **PostgreSQL:** O banco de dados está na porta **5432**. Ele só é acessível pelos outros containers. Para interagir com ele, você pode usar uma ferramenta de banco de dados ou entrar no container com **docker exec -it postgres_container psql -U admin -d meubanco**.

Etapa 1: Estrutura do Projeto - Infraestrutura Base (Docker, Python, Postgres)

1º Passo: Criei o arquivo `docker-compose.yml`, responsável para iniciar a API, banco de dados e o n8n. A primeira instrução foi o banco de dados, contendo uma arquitetura simples apenas para suportar os dados disponibilizados.

Estrutura do banco e tabela:

Criei um banco de dados relacional no PostgreSQL para centralizar os dados de eventos que antes estavam dispersos em planilhas, para isso, utilizei o comando: **docker-compose exec postgres_container psql -U admin -d meubanco**.

Dentro dele, criei a tabela **eventos**, que armazena informações de cada evento de forma organizada e padronizada. Usei uma tabela única, porque todos os dados possuem o mesmo contexto (eventos).

Estrutura da tabela:

```
meubanco=# \d eventos
```

Column	Type	Collation	Nullable	Default
id	integer		not null	nextval('eventos_id_seq'::regclass)
nome_evento	text		not null	
data_anual	text			
descricao	text			
alcance	integer			0
engajamento	integer			0
status	text			
fonte	text			

Indexes:

```
"eventos_pkey" PRIMARY KEY, btree (id)
```

A tabela foi criada de forma flexível para garantir que todos os dados vindos das planilhas integradas pelo n8n sejam inseridos no banco sem falhas, priorizando a demonstração do fluxo automatizado em vez da validação rígida dos dados.

2º Passo: Criei o arquivo **main.py** que contém o código principal da API, construída com o framework **FastAPI**.

Nele, foram definidos os seguintes pontos:

- Conexão com o banco de dados PostgreSQL, utilizando a biblioteca `psycopg2`;

- Definição do modelo de dados Evento com Pydantic, que garante validação dos campos recebidos;
- Implementação dos endpoints principais:
 - GET / - rota de teste da API;
 - POST /eventos/ - cria um novo evento;
 - GET /eventos/ - lista todos os eventos cadastrados;
 - GET /eventos/{nome_evento} - consulta um evento específico;
 - PUT /eventos/{nome_evento} - atualiza um evento;
 - DELETE /eventos/{nome_evento} - apaga um evento.

Esse arquivo representa a camada de acesso aos dados, permitindo que a aplicação insira, consulte, atualize e remova registros no banco.

3º Passo: Criei o arquivo **requirements.txt** para conter a lista de dependências necessárias, garantindo que a API funcione corretamente. O Docker utiliza esse arquivo para instalar automaticamente todas as bibliotecas no ambiente do container.

Inclui as bibliotecas:

- fastapi → framework web utilizado.
- uvicorn[standard] → servidor ASGI para rodar a aplicação.
- psycopg2-binary → driver para conectar ao banco PostgreSQL.

4º Passo: Criei o arquivo **Dockerfile** que define como a imagem da API deve ser construída. Esse arquivo garante que a API seja reproduzida em qualquer ambiente, de forma padronizada.

Principais etapas configuradas:

- Baseado na imagem oficial do Python 3.9 slim;
- Configuração do diretório de trabalho /app;
- Instalação das dependências a partir do requirements.txt;
- Cópia do código da API para dentro da imagem;
- Exposição da porta **8000**, utilizada pelo FastAPI;
- Comando para iniciar a aplicação com Uvicorn.

5º Passo: Atualizei o arquivo **docker-compose.yml** para orquestrar os serviços. Nessa etapa, são iniciados dois containers, garantindo que o banco de dados e a API sejam executados de forma integrada, o que simplifica o gerenciamento:

1. Postgres

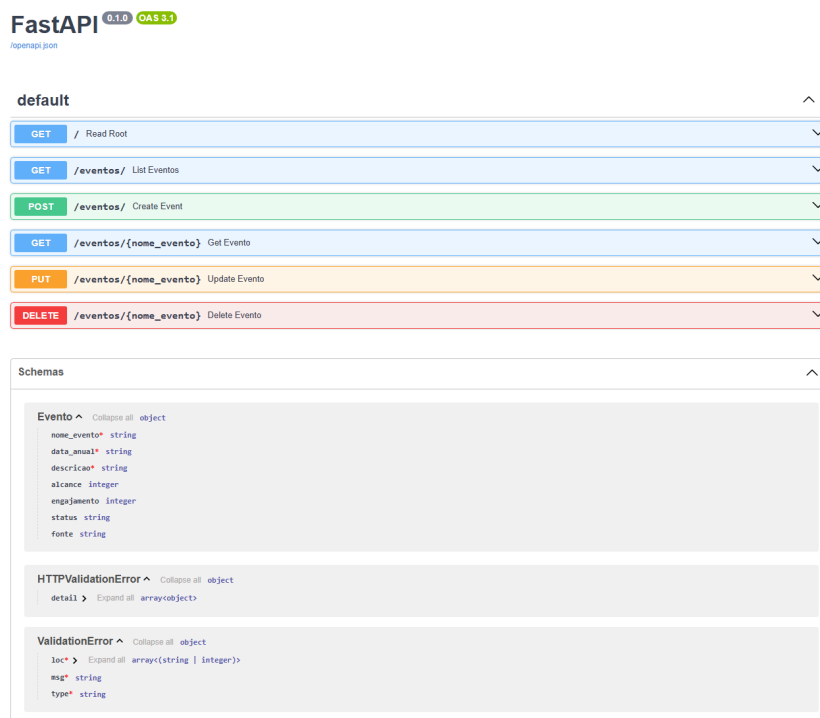
- Banco de dados relacional para armazenar os eventos;
- Configurado com usuário, senha e nome do banco via variáveis de ambiente;
- Persistência dos dados através de volume mapeado.

2. API (FastAPI)

- Contêiner construído a partir do Dockerfile;
- Conectado ao banco de dados via variáveis de ambiente;
- Porta 8000 exposta para acessar os endpoints da API;
- Dependente do serviço do banco (depends_on).

A imagem abaixo mostra a documentação automática do FastAPI:

Acessando pelo link <http://localhost:8000> aparece `{"status": "API está funcionando!"}` e pelo <http://localhost:8000/docs> aparece a documentação automática do FastAPI:



Etapa 2: Workflow de Migração de Dados (ETL com n8n):

1º Passo: Atualizei o arquivo docker-compose.yml adicionando o n8n, que agora inclui os três serviços (PostgreSQL, API e n8n), permitindo que eles se comuniquem. Após a execução do arquivo docker-compose.yml com comando docker-compose up -d --build, acessei o endereço <http://localhost:5678> com login e senha que criei, começando a usar o workflow para a migração de dados.

2º Passo: Estabeleci um fluxo de trabalho no n8n e o processo começou de forma manual, executando o fluxo. Criei uma conta no Google Sheets e conectei ao n8n. Autorizei o acesso com minhas credenciais seguindo o passo a passo que o próprio n8n mostrou. A próxima etapa foi configurar a leitura de três planilhas distintas usando nós Google Sheets. Cada planilha contém informações que precisei unificar em uma única estrutura. Para garantir a consistência dos dados, criei etapas intermediárias de edição de campos, onde padronizei colunas, ajustei formatos (como datas e textos) e corriji divergências entre as diferentes fontes.

Ou seja:

- Criei um fluxo de trabalho que começa manualmente, quando executo o fluxo;
- Configurei uma conta no **Google Sheets** e conectei ao **n8n**;
- Autorizei o acesso com minhas **credenciais**, seguindo o passo a passo do próprio **n8n**;
- Configurei a leitura de **três planilhas** distintas usando nós do **Google Sheets**;
- Cada planilha continha informações que eu precisava unificar em uma **estrutura única**;
- Para garantir a consistência dos dados, criei etapas intermediárias de edição de campos, onde:
 - Padronizei colunas;
 - Ajustei formatos (como datas e textos);
 - Corrigi divergências entre as diferentes fontes.

Conta do Planilhas Google

API OAuth2 do Planilhas Google

Conexão

Conta conectada

Reconectar: Sign in with Google

Compartilhamento

Detalhes

Precisa de ajuda para preencher estes campos? Abra os documentos

URL de redirecionamento OAuth

http://localhost:5678/rest/oauth2-credential/callback

No Planilhas Google, use o URL acima quando solicitado a inserir um retorno de chamada OAuth ou um URL de redirecionamento

ID do cliente *

301333564758-fmbad9f6e8uu502po20ku2ncska0n.apps.googleusercontent.com

Segredo do cliente *

Certifique-se de ter habilitado as seguintes APIs e serviços no Console do Google Cloud: API do Google Drive, API do Planilhas Google. Saiba mais.

Usuários do plano Enterprise podem obter credenciais de cofres externos. Mais informações

ENTRADA

Get row(s) in sheet

3 itens

Editar Campos1

Executar passo

Parâmetros

Configurações

Doc

Modo

Manual Mapping

Campos a definir

fonte

String

IA

Arraste os campos de entrada aqui

Incluir outros campos de entrada

Campos de entrada a serem incluídos

Opções

Adicionar opção

SAÍDA

3 itens

número_da_linha	Dados/Anual	Evento	Descrição	Engajamento	STATUS	fonte
2	01/05/2025 a 31/07/2025	Evento IA Facit	Apresentação de slides	925	Concluído	IA
3	01/05/2025 a 31/07/2025	Conexão Facit/IA	Evento de networking para profissionais	890	Concluído	IA
4	01/05/2025 a 31/07/2025	Palestras Facit/IA	Palestras didáticas	150	Concluído	IA

ENTRADA

Get row(s) in sheet

9 itens

Editar Cam...

Executar passo

Parâmetros

Configurações

Doc

Modo

Manual Mapping

Campos a definir

fonte

String

MKT

Arraste os campos de entrada aqui

Incluir outros campos de entrada

Campos de entrada a serem incluídos

Opções

Adicionar opção

SAÍDA

9 itens

número_da_linha	Dados/Anual	Evento	Descrição	STATUS	fonte
2	23/06/2022 a 30/06/2022	Campanha "Conecte que Conecta"	Iniciativa para aumentar o engajamento com a comunidade	Concluído	MKT
3	03/05/2025	Site Facit	Integração do site institucional	Concluído	MKT
4	01/01/2025 a 31/03/2025 (Trimestre)	Análise de Desempenho e Metas Q1	Reunião para avaliar as estatísticas do primeiro trimestre.	Concluído	MKT
5	01/08/2025 a 31/10/2025 (Semestral)	Ortização de Anúncios Pago	Estratégia interna focada em promoção de campanhas de tráfego pago	Em andamento	MKT
6	Evento	Evento	Evento	Concluído	MKT
7	01/05/2025 a 31/07/2025	Planejamento Estratégico 2026	Fruituação dos projetos de 2025 e criação do plano estratégico para 2026	Concluído	MKT
8	01/05/2025 a 31/06/2025 (Trimestre)	Análise de Desempenho e Metas Q1	Reunião para avaliar as estatísticas do primeiro trimestre.	Concluído	MKT
9	01/07/2025 a 31/10/2025 (Trimestre)	Análise de Desempenho e Metas Q1	Reunião para avaliar as estatísticas do primeiro trimestre.	Em andamento	MKT

ENTRADA

Get row(s) in sheet2

6 itens

Editar Cam...

Executar passo

Parâmetros

Configurações

Doc

Modo

Manual Mapping

Campos a definir

fonte

String

RH

Arraste os campos de entrada aqui

Incluir outros campos de entrada

Campos de entrada a serem incluídos

Opções

Adicionar opção

SAÍDA

6 itens

número_da_linha	Dados/Anual	Evento	Descrição	ALCANCE	STATUS	fonte
2	10/05/2025 a 10/05/2025	Facilita Comunica	Apresentação de Casos de Sucesso	200	Não iniciado	RH
3	01/05/2025 a 31/07/2025	Facilita em Ação: Multiplicando conhecimento para nossos colaboradores	Campanha EDUCATIVA	1055	Concluído	RH
4	01/01/2025 a 31/10/2025	Gente conectada	Conexão/tráfego pago entre colaboradores	465	Em andamento	RH
5	01/05/2025 a 31/08/2025	Facilidade de Recrutamento Inteligente	Plano de AÇÃO	150	Em andamento	RH
6	10/06/2025 a 31/06/2025	Pesquisa de satisfação	Campanha último lugar para trabalhar	500	Não iniciado	RH
7	04/05/2025	RH Agil com Facit	Descomplicando a Gestão de Pessoas	430	Concluído	RH

3º Passo: Após esse ajuste, implementei um nó de **mesclagem** para consolidar os registros em um único conjunto de dados. Dessa forma, os itens provenientes das **três planilhas** foram reunidos em uma coleção final, eliminando redundâncias e preparando tudo para a carga no banco.

The screenshot shows the 'Mesclar' node configuration. The 'Entrada' section displays three JSON items being merged. The 'Saída' section shows the resulting merged data table with 17 items.

nome_evento	dados_anuais	descricao	engajamento	alcance	status	fonte
Desafio IA Facilit	01/05/2025 a 31/07/2025	Apresentação de ideias	920	0	Concluído	IA
Conexão Facilit IA	01/05/2025 a 31/07/2025	Evento de networking para profissionais	890	0	Concluído	IA
Palestras Facilit AI	01/05/2025 a 31/07/2025	Palestras dinâmicas	1150	0	Concluído	IA
Campaña "Conectado que Conecta"	23/06/2022 a 30/06/2022	Iniciativa para aumentar o engajamento com a comunidade	0	0	Concluído	MKT
Wiki Facilit	02/05/2025	Inauguração do site institucional	0	0	Concluído	MKT
Análise de Desempenho e Metas Q1	01/01/2025 a 31/03/2025 (Trimestre)	Reunião para avaliar as estatísticas do primeiro trimestre	0	0	Concluído	MKT
Optimização de Anúncios Pagos	01/08/2025 a 31/07/2025 (Semestral)	Estratégia interna focada em promoção de campanhas de tráfego pago	0	0	Em andamento	MKT

4º Passo: Para refinar ainda mais as informações, adicionei uma etapa de **edição de campos**, seguida por um nó **code** customizado, que foi utilizado para aplicar regras adicionais, como a geração de identificadores e a validação de valores obrigatórios.

The screenshot shows the 'Editar campos' node configuration. The 'Entrada' section displays the merged data. The 'Saída' section shows the data after field editing, with some values updated.

nome_evento	dados_anuais	descricao	engajamento	alcance	status	fonte
Desafio IA Facilit	01/05/2025 a 31/07/2025	Apresentação de ideias	920	0	Concluído	IA
Conexão Facilit IA	01/05/2025 a 31/07/2025	Evento de networking para profissionais	890	0	Concluído	IA
Palestras Facilit AI	01/05/2025 a 31/07/2025	Palestras dinâmicas	1150	0	Concluído	IA
Campaña "Conectado que Conecta"	23/06/2022 a 30/06/2022	Iniciativa para aumentar o engajamento com a comunidade	0	0	Concluído	MKT
Wiki Facilit	02/05/2025	Inauguração do site institucional	0	0	Concluído	MKT
Análise de Desempenho e Metas Q1	01/01/2025 a 31/03/2025 (Trimestre)	Reunião para avaliar as estatísticas do primeiro trimestre	0	0	Concluído	MKT
Optimização de Anúncios Pagos	01/08/2025 a 31/07/2025 (Semestral)	Estratégia interna focada em promoção de campanhas de tráfego pago	0	0	Em andamento	MKT

The screenshot shows the 'Código' node configuration. The 'Entrada' section displays the merged data. The 'Saída' section shows the data after code execution, with some values updated.

nome_evento	dados_anuais	descricao	engajamento	alcance	status	fonte
Desafio IA Facilit	01/05/2025 a 31/07/2025	Apresentação de ideias	920	0	Concluído	IA
Conexão Facilit IA	01/05/2025 a 31/07/2025	Evento de networking para profissionais	890	0	Concluído	IA
Palestras Facilit AI	01/05/2025 a 31/07/2025	Palestras dinâmicas	1150	0	Concluído	IA
Campaña "Conectado que Conecta"	23/06/2022 a 30/06/2022	Iniciativa para aumentar o engajamento com a comunidade	0	0	Concluído	MKT
Wiki Facilit	02/05/2025	Inauguração do site institucional	0	0	Concluído	MKT
Análise de Desempenho e Metas Q1	01/01/2025 a 31/03/2025 (Trimestre)	Reunião para avaliar as estatísticas do primeiro trimestre	0	0	Concluído	MKT
Optimização de Anúncios Pagos	01/08/2025 a 31/07/2025 (Semestral)	Estratégia interna focada em promoção de campanhas de tráfego pago	0	0	Em andamento	MKT
Planejamento Estratégico 2026	01/05/2025 a 31/07/2025	Finalização dos projetos de 2025 e início do plano estratégico para 2026	0	0	Concluído	MKT
Análise de Desempenho e Metas Q1	01/01/2025 a 31/03/2025 (Trimestre)	Reunião para avaliar as estatísticas do primeiro trimestre	0	0	Concluído	MKT

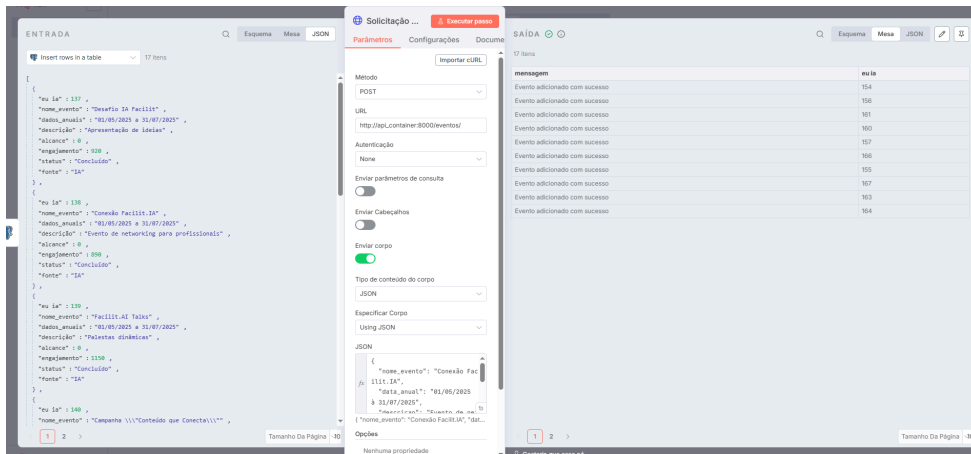
5º Passo: Com os dados devidamente tratados, configurei a etapa de inserção em uma tabela do **PostgreSQL**. Criei a credencial do Postgres no n8n, conectando o banco chamado meubanco. Para isso, informei o host do serviço, o **usuário** e a **senha** configurados no **docker-compose.yml**. Após isso, testei a conexão e validei que estava funcionando. Nessa fase, os registros já entram padronizados e estruturados conforme o modelo que defini anteriormente no banco.

The screenshot shows the 'Conta Postgres 2' configuration window in n8n. The 'Conexão' tab is active, showing a green status bar that says 'Conexão testada com sucesso' (Connection tested successfully) with a 'Retry' button. Below this, the 'Detalhes' (Details) section contains the following fields: 'Hospedar' (Host) set to 'postgres', 'Banco de dados' (Database) set to 'meubanco', 'Usuário' (User) set to 'admin@le', and 'Senha' (Password) masked with dots. The 'Número máximo de conexões' (Maximum number of connections) is set to '200'. A message at the top of the details section says 'Precisa de ajuda para preencher estes campos? Abra os documentos' (Need help filling these fields? Open the documents).

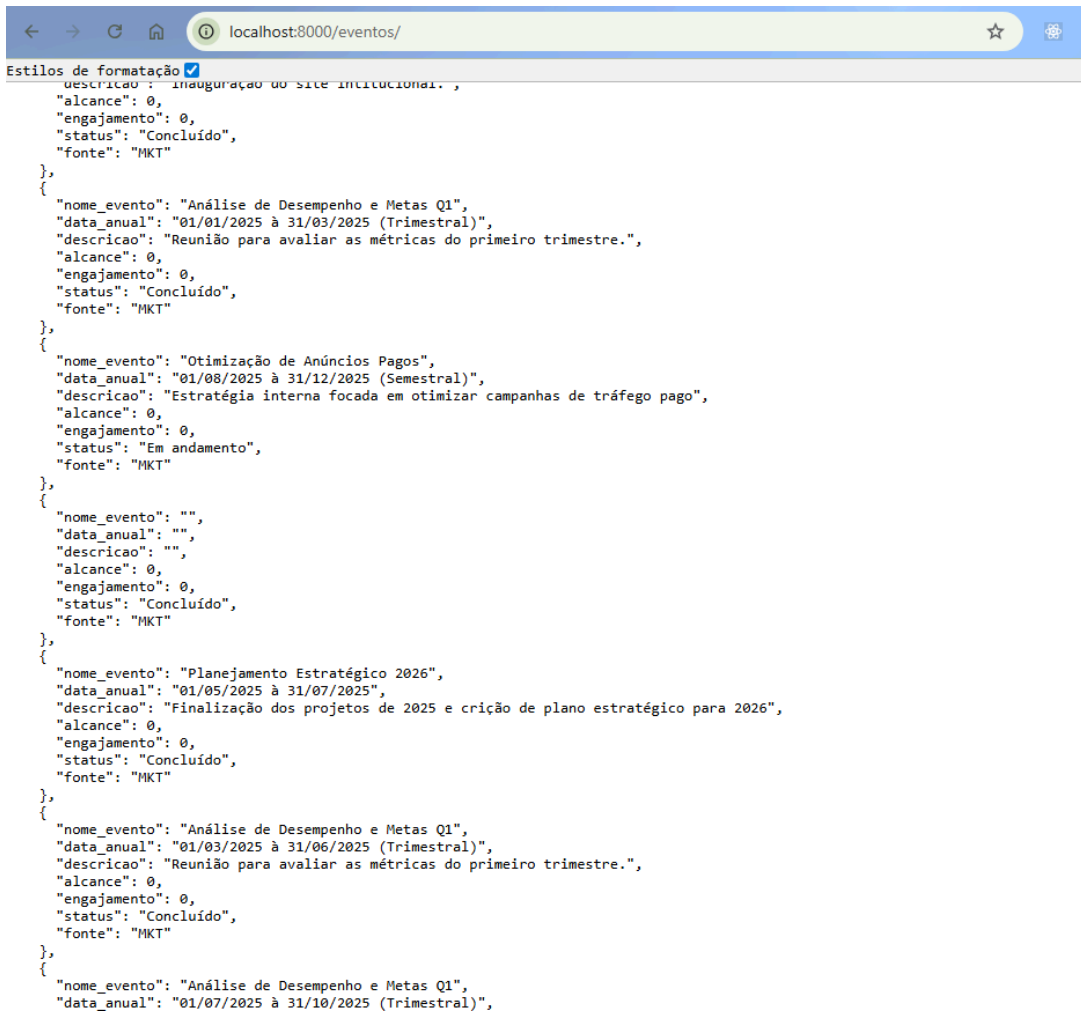
The screenshot shows the n8n interface with two panels. The left panel, titled 'ENTRADA' (Input), shows a JSON array of 17 items. The right panel, titled 'SAÍDA' (Output), shows a table with 17 rows of data. The table has columns: 'id', 'nome_evento', 'dados_anual', 'descricao', 'alcance', 'engajamento', 'status', and 'fonte'. The data is as follows:

id	nome_evento	dados_anual	descricao	alcance	engajamento	status	fonte
137	Desafio IA Facilit	01/05/2025 a 31/07/2025	Apresentação de ideias	0	826	Concluído	IA
138	Conexão Facilit IA	01/05/2025 a 31/07/2025	Evento de networking para profissionais	0	890	Concluído	IA
139	Palestras Facilit IA	01/05/2025 a 31/07/2025	Palestras dinâmicas	0	1150	Concluído	IA
140	Campaña ('Conteúdo que Conecta')	23/06/2022 a 30/06/2022	Iniciativa para aumentar o engajamento com a comunidade	0	0	Concluído	MKT
141	Wiki Facilit	02/05/2025	Inauguração do site institucional.	0	0	Concluído	MKT
142	Análise de Desempenho e Metas Q1	01/01/2025 a 31/03/2025 (Trimestre)	Revisão para avaliar as estatísticas do primeiro trimestre.	0	0	Concluído	MKT
143	Otimização de Anúncios Pagos	01/08/2025 a 31/12/2025 (Semestral)	Estratégia interna focada em promoção de campanhas de tráfego pago.	0	0	Em andamento	MKT
144	[vazio]	[vazio]	[vazio]	0	0	Concluído	MKT
145	Planejamento Estratégico 2026	01/05/2025 a 31/07/2025	Finalização dos projetos de 2025 e início do plano estratégico para 2026.	0	0	Concluído	MKT
146	Análise de Desempenho e Metas Q1	01/01/2025 a 31/03/2025 (Trimestre)	Revisão para avaliar as estatísticas do primeiro trimestre.	0	0	Concluído	MKT

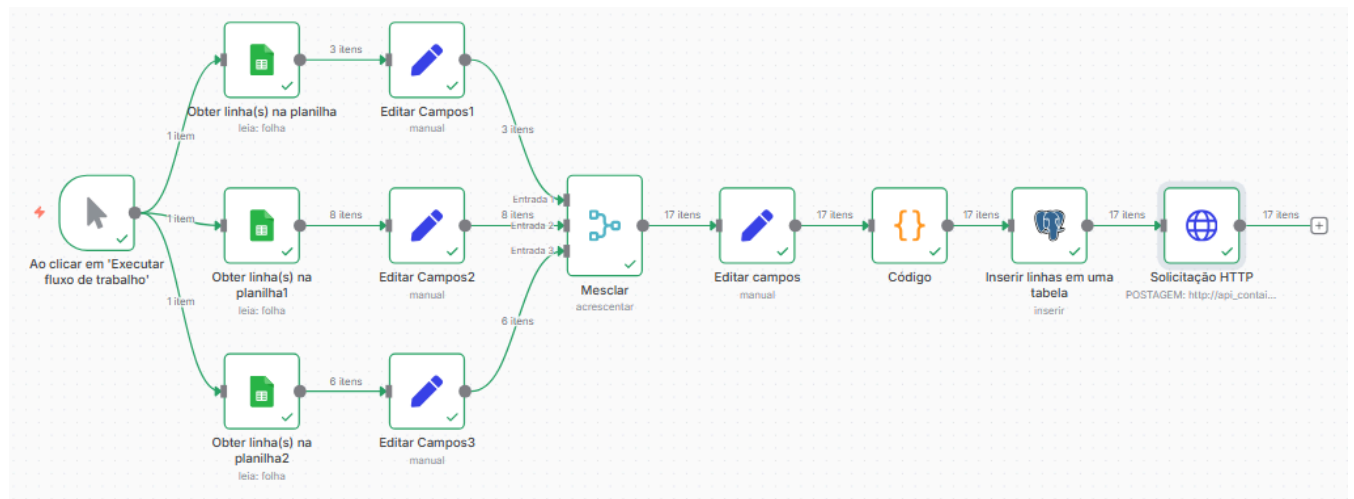
6º Passo: Por fim, adicionei uma chamada **HTTP** do tipo **POST** para a **API** que construí em FastAPI. Essa etapa permite que os dados inseridos no banco também sejam transmitidos para a API, servindo tanto para validação como para a integração com outras camadas da solução.



e quando acessei endereço <http://localhost:8000/ eventos/> mostrou os dados salvos no banco de dados .



Arquitetura do fluxo criado para a segunda etapa do desafio:



Etapa 3: Workflow de Consulta com IA (O Cérebro do Projeto):

1º Passo: Acessei o site <https://webhook.site/> e automaticamente foi gerada uma URL exclusiva. Copiei essa URL e usei em um sistema que envia requisições HTTP. Assim que os dados foram enviados, visualizei tudo na tela.

<https://webhook.site/#!/view/c09e4dff-7a8b-4aab-9f0f-74bc62ab58af/63c90750-2510-403e-a20-bca45ecddee0/1>

Request Details & Headers

Field	Value
Host	186.233.111.168
Date	20/08/2025 18:55:50 (há 3 dias)
Size	215 bytes
Time	0.000 sec
ID	63c90750-2510-403e-a20-bca45ecddee0
Note	

Request Content

```
{
  "row_number": 3,
  "Data/Anual": "01/08/2025 à 31/07/2025",
  "Evento": "Facilita a ação: Multiplicando conhecimento para os nossos colaboradores.",
  "Descrição": "Campanha EDUCATIVA",
  "ALCANCE": 1000,
  "STATUS": "Concluído"
}
```

2º Passo: Criei um novo workflow no n8n e comecei adicionando uma URL teste:

<https://webhook.site/c09e4dff-7a8b-4aab-9f0f-74bc62ab58af>

Webhook3

Ouçá o evento de teste

ParâmetrosConfiguraçõesDocumentos

URLs de webhook

URL de testeURL de produção

Public

http://localhost:5678/webhook-test/consulta-ia

Método HTTP

POST

Caminho

consulta-ia

Autenticação

None

Responder

Streaming

Insira um nó que suporte streaming (por exemplo, "Agente de IA") e habilite o streaming para transmitir diretamente para a resposta enquanto o fluxo de trabalho é executado. [Mais detalhes](#)

Opções

Nenhuma propriedade

Adicionar opção

3º Passo: Usei o Postman como ponto de entrada da pergunta do usuário:

http://localhost:5678/webhook-test/consulta-ia

SaveShare

POSThttp://localhost:5678/webhook-test/consulta-iaSend

ParamsAuthorizationHeaders (8)BodyScriptsTestsSettings

noneform-datax-www-form-urlencodedrawbinaryGraphQLJSON

1 { "pergunta": "Qual o período do evento Conexão Facilit.IA?" }

2

BodyCookiesHeaders (7)Test Results

JSONPreviewVisualize

200 OK428 ms226 B

Webhook3

Ouçá o evento de teste

ParâmetrosConfiguraçõesDocumentos

URLs de webhook

URL de testeURL de produção

Public

http://localhost:5678/webhook-test/consulta-ia

Método HTTP

POST

Caminho

consulta-ia

Autenticação

None

Responder

Streaming

Insira um nó que suporte streaming (por exemplo, "Agente de IA") e habilite o streaming para transmitir diretamente para a resposta enquanto o fluxo de trabalho é executado. [Mais detalhes](#)

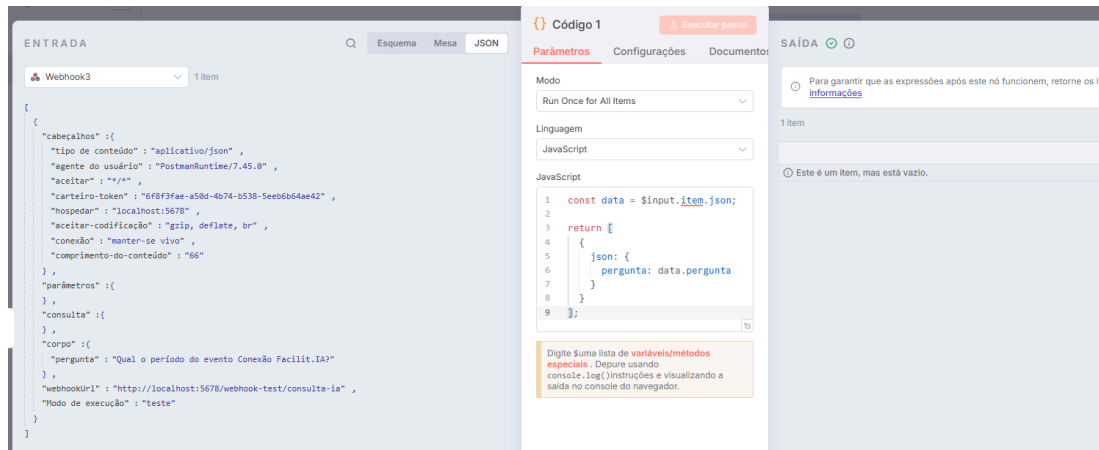
SAÍDA

EsquemaMesaJSON

1 item

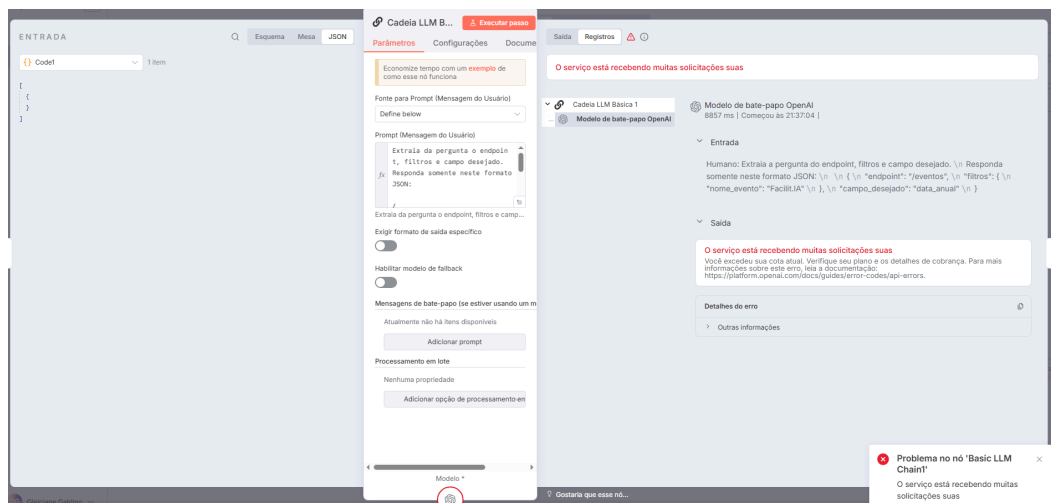
cabeçalhos	parâmetros	consulta	corpo	URL do webhook	modo de execução
<div>tipo de conteúdo : aplicativo/json</div> <div>agente do usuário : PostmanRuntime/7.45.0</div> <div>aceitar : */*</div> <div>token do cartão : 7209ef98-5627-4603-8d6e-0071ee500d89</div> <div>anfitrião : localhost:5678</div> <div>aceitar-codificação : gzip, deflate, br</div> <div>conexão : keep-alive</div> <div>comprimento do conteúdo : 66</div>	[objeto vazio]	[objeto vazio]	pergunta : Qual o período do evento Conexão Facilit.IA?	http://localhost:5678/webhook-test/consulta-ia	teste

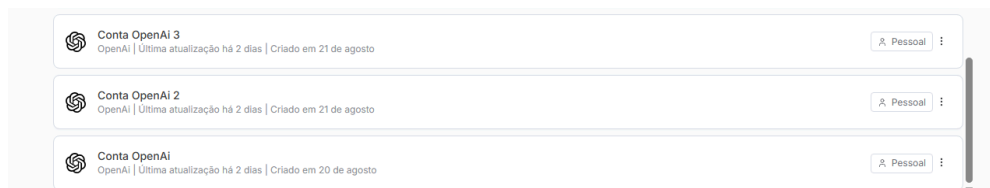
4º Passo: Adicionei um nó **code** apenas para extrair o campo pergunta da requisição e passa para frente no formato **JSON**:



5º Passo: Adicionei um nó **Basic LLM Chain1** (junto a OpenAI Chat Model), com a função de receber a pergunta e transformar em instruções estruturadas no formato **JSON**. Ou seja, ele entende qual **endpoint da API** chamar, quais filtros aplicar e qual campo buscar.

```
{  
  "endpoint": "/eventos",  
  "filtros": { "nome_evento": "Facilita" },  
  "campo_desejado": "data_anual"  
}
```

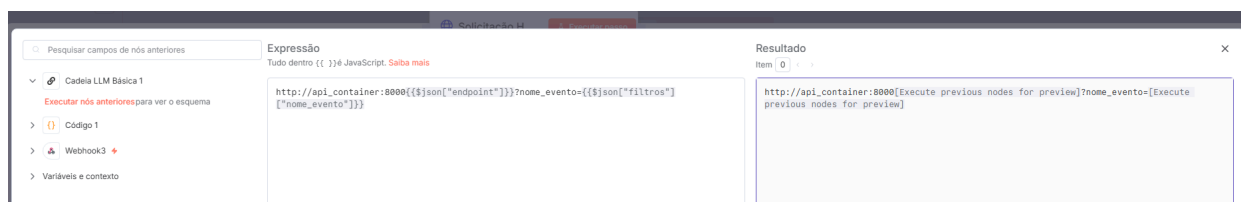




(para adicionar na credencial do n8n, criei algumas chaves de testes, além da que foi disponibilizada no desafio)

6º Passo: Criei um nó **request** para montar a URL dinamicamente usando as variáveis do JSON do LLM:

http://api_container:8000/eventos?nome_evento=Facilita.IA



7º Passo: Adicionei um novo nó **Basic LLM Chain2** (com outra OpenAI Chat Model) para receber a pergunta original e os dados crus da API. A ideia é usar o LLM como "pós-processador", gerando um JSON de resposta mais natural e direto.



8º Passo: Criei um novo **code** para implementar uma lógica extra de validação, onde ele lê o JSON retornado da API. A ideia é verificar os campos desejados e o nome do evento que o LLM extraiu. Com isso, vai procurar o evento na lista da API e retorna apenas o valor específico ou "Evento não encontrado".

Exemplo:

```
{
  "evento": "Facilita.IA",
  "campo": "data_anual",
```

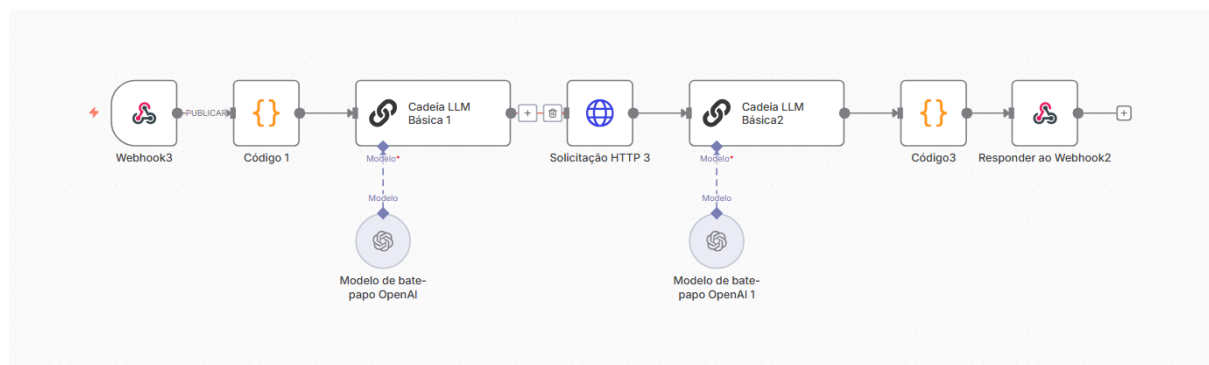
```
"resposta": "01/05/2025 à 31/07/2025"
}
```

9º Passo: O último nó adicionado foi o **Respond to Webhook**, retornando o JSON final ao cliente que chamou o webhook.

Exemplo de resposta:

```
{
  "evento": "Facilit.IA",
  "campo": "data_anual",
  "resposta": "01/05/2025 à 31/07/2025"
}
```

Arquitetura do fluxo criado para a terceira etapa do desafio:



Desafios encontrados:

Um dos maiores desafios que enfrentei neste projeto foi o primeiro contato com o n8n e a OpenAI. Nunca tinha trabalhado com automação de fluxo de dados ou com inteligência artificial, então tudo era novo e, muitas vezes, parecia um grande quebra-cabeça. A princípio, entender como o n8n conectava as planilhas do Google Sheets à minha API e depois ao banco de dados parecia bem complicado. A cada erro de JSON inválido ou de variáveis que não eram reconhecidas, a sensação de estar começando do zero voltava. No entanto, a cada problema resolvido, a cada nó que funcionava como o esperado, o processo se tornava incrivelmente gratificante. No final, percebi que por mais desafiador que tenha sido, o conhecimento que adquiri sobre ETL e integração de sistemas foi enorme. Foi uma jornada de aprendizado que me mostrou o poder da automação para simplificar tarefas complexas e que, com certeza, levarei para meus próximos projetos.